Hepatoprotective effects of systemic ER activation

ER regulated genes and NAFLD classification models

Christian Sommerauer & Carlos Gallardo

19 September, 2022

```
# source and library import
source('code/00_helper_functions.R')
library(tidyverse)
library(RColorBrewer)
library(ComplexHeatmap)
library(caret)
library(multiROC)
library(patchwork)
library(ggpubr)
```

```
# color palettes
colPals <- list()
colPals$conditions <- setNames(c('#E98BB6', '#B02262', '#7F9AD7', '#2A2F72', '#7DC7D1', '#339ACD', '#35
                               c('CDf', 'HFDf', 'CDm', 'HFDm', 'DPN', 'DIP', 'E2', 'PPT'))
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
colPals$clusters <- setNames(c('#A9D265', '#82506D', '#FA9F1C', '#676A6E'),
                             c('1', '2', '3', '4'))
colPals$celltypes <- setNames(c('#B4272F', '#E5462D', '#FFD1D1', '#F4E54C', '#FBAA3E', '#AA654E', '#B58
                                '#B3177E', '#CAC1DD', '#67227D','#36B449', '#82C349', '#A9D265', '#1994
                              c('Cholangiocytes', 'Endothelial cells', 'HsPCs', 'Stromal cells', 'Hepat
                                'Kupffer cells', 'Monocytes & Monocyte-derived cells', 'T cells', 'NK c
                                'ILC1s', 'B cells', 'cDC1s', 'cDC2s', 'Mig. cDCs', 'pDCs', 'Basophils',
colPals$inferno <- c('#FCFFA4', '#FCA50A', '#DD513A', '#932667', '#420A68', '#000004')
colPals$stage <- setNames(c('#F6F6F6','#FAD7A0','#7B241C'),
                          c('CTRL','NAFL','NASH'))
colPals$NAS <- setNames(c('#F6F6F6','#BFC9CA','#000000'),
                        c('low','mid','high'))
colPals$fibrosis <- setNames(c('#F6F6F6','#DF7B71','#000000'),
                             c('low','mid','high'))
```

Load data

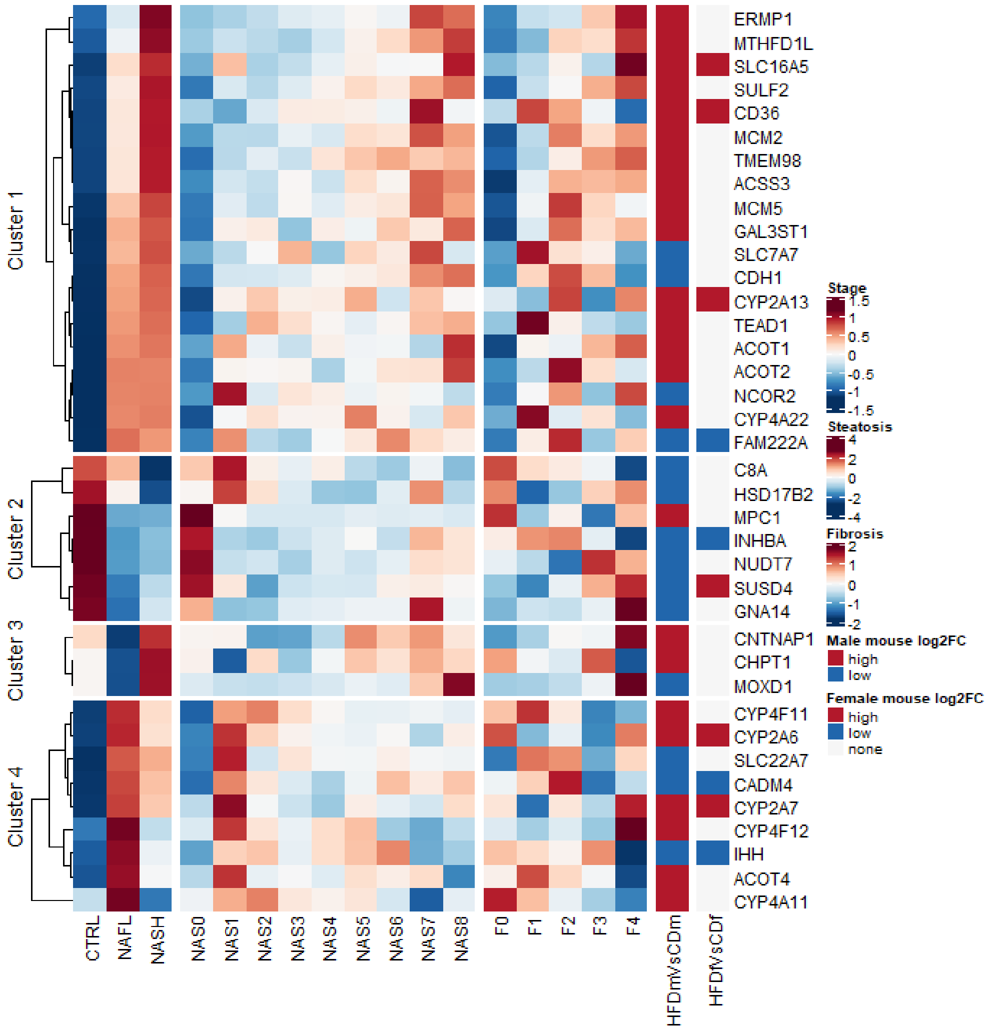
```
# mouse-human orthologs
mouse_human_orthologs <- read.table(
  file = 'data/ensembl_mmus_hsap_sep2019_orthologs.tsv',
  sep = '\t',
  header = TRUE,
  quote = '')
```

```r
# consensus differentially expressed genes
DEGs <- readRDS('results/bulkRNAseq_mmus_DEGs.rds')

# RNAseq data
RNAseq <- readRDS('results/bulkRNAseq_mmus_data_filt_norm.rds')

# human cohort data from:
# Govaere et al. 2020 (https://doi.org/10.1126/scitranslmed.aba4448)
# Kozumi et al. 2021 (https://doi.org/10.1002/hep.31995)
# GTEx v8 (https://www.gtexportal.org/home/datasets)
cohort_data <- readRDS('data/bulkRNAseq_human_cohort_data.rds')

cohort_data$Govaere$cpm_filt <- cohort_data$Govaere$cpm %>%
  tibble::rownames_to_column(var = 'gene') %>%
  dplyr::filter(gene %in% mouse_human_orthologs$GeneID_human) %>%
  dplyr::mutate(gene = dplyr::recode(gene, !!!setNames(mouse_human_orthologs$GeneSymbol_human,
                                                       mouse_human_orthologs$GeneID_human))) %>%
  dplyr::filter(!duplicated(gene) & gene != "") %>%
  tibble::column_to_rownames(var = 'gene')

cohort_data$Kozumi$cpm_filt <- cohort_data$Kozumi$cpm %>%
  dplyr::filter(rownames(.) %in% mouse_human_orthologs$GeneSymbol_human
                & !duplicated(rownames(.)))

cohort_data$GTEx_liver_CTRL$cpm_filt <- cohort_data$GTEx_liver_CTRL$cpm %>%
  dplyr::filter(rownames(.) %in% mouse_human_orthologs$GeneSymbol_human
                & !duplicated(rownames(.)))

# gene sets
gene_sets <- list()
gene_sets[['ER_genes']] <- scan('results/ER_regulated_genes.txt', what = character(), quiet = T)
gene_sets[['NAS_markers']] <- scan('results/ER_regulated_genes_NAS_markers.txt', what = character(), qu
gene_sets[['fibrosis_markers']] <- scan('results/ER_regulated_genes_fibrosis_markers.txt', what = charac
gene_sets[['sulf2']] <- c('SULF2')
gene_sets[['thbs2']] <- c('THBS2')
gene_sets[['combined_sulf2_thbs2']] <- c('SULF2','THBS2')
```

Integrated heatmap of ER regulated genes across human NAFLD spectrum

```r
human_stage <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% gene_sets$ER_genes) %>%
  groupTransform(cohort_data$Govaere$meta$Stage, function(x) apply(x,1,median)) %>%
  scaleData(method = 'zscore') %>%
  dplyr::arrange(rownames(.))

human_steatosis <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% gene_sets$ER_genes) %>%
  groupTransform(cohort_data$Govaere$meta$NAS, function(x) apply(x,1,median)) %>%
  scaleData(method = 'zscore') %>%
  dplyr::arrange(rownames(.))
```

```
human_fibrosis <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% gene_sets$ER_genes) %>%
  groupTransform(cohort_data$Govaere$meta$Fibrosis, function(x) apply(x,1,median)) %>%
  scaleData(method = 'zscore') %>%
  dplyr::arrange(rownames(.))

mouse_log2FC_HFDmVsCDm <- DEGs$filt$CDmVsHFDm %>%
  dplyr::rename(GeneSymbol_mouse = external_gene_name) %>%
  dplyr::inner_join(mouse_human_orthologs, by = 'GeneSymbol_mouse') %>%
  dplyr::filter(GeneSymbol_human %in% gene_sets$ER_genes) %>%
  dplyr::filter(!duplicated(GeneSymbol_human)) %>%
  dplyr::select(GeneSymbol_human, log2FoldChange) %>%
  dplyr::mutate(log2FoldChange = log2FoldChange*-1) %>%
  dplyr::rename(HFDmVsCDm = log2FoldChange) %>%
  dplyr::mutate(HFDmVsCDm = factor(ifelse(HFDmVsCDm>0, 'high', 'low'), levels = c('high','low'))) %>%
  tibble::column_to_rownames(var = 'GeneSymbol_human') %>%
  dplyr::arrange(rownames(.))

mouse_log2FC_HFDfVsCDf <- DEGs$filt$CDfVsHFDf %>%
  dplyr::rename(GeneSymbol_mouse = external_gene_name) %>%
  dplyr::inner_join(mouse_human_orthologs, by = 'GeneSymbol_mouse') %>%
  dplyr::filter(GeneSymbol_human %in% gene_sets$ER_genes) %>%
  dplyr::filter(!duplicated(GeneSymbol_human)) %>%
  dplyr::select(GeneSymbol_human, log2FoldChange) %>%
  dplyr::mutate(log2FoldChange = log2FoldChange*-1) %>%
  dplyr::rename(HFDfVsCDf = log2FoldChange) %>%
  dplyr::mutate(HFDfVsCDf = ifelse(HFDfVsCDf>0, 'high', 'low')) %>%
  dplyr::add_row(GeneSymbol_human = dplyr::setdiff(rownames(mouse_log2FC_HFDmVsCDm),
                                                   .$GeneSymbol_human),
                 HFDfVsCDf = 'none') %>%
  dplyr::mutate(HFDfVsCDf = factor(HFDfVsCDf, levels = c('high','low', 'none'))) %>%
  tibble::column_to_rownames(var = 'GeneSymbol_human') %>%
  dplyr::arrange(rownames(.))

set.seed(4)
clusters_ER_genes <- kmeans(human_stage, centers = 4, iter.max = 100)

Heatmap(human_stage, width = unit(24, "mm"),name = "Stage",
        split = clusters_ER_genes$cluster,
        col = circlize::colorRamp2(breaks=seq(-max(abs(human_stage)), max(abs(human_stage)), length.out=
                                   colors=colPals$RdBu),
        row_title = c("Cluster 1", "Cluster 2", "Cluster 3", "Cluster 4"),
        cluster_row_slices = FALSE,
        column_order=c("CTRL", "NAFL", "NASH")) +
  Heatmap(human_steatosis, width = unit(72, "mm"),name = "Steatosis",
          column_order=paste0('NAS', seq(0,8)),
          col = circlize::colorRamp2(breaks=seq(-max(abs(human_steatosis)), max(abs(human_steatosis)), l
                                     colors=colPals$RdBu)) +
  Heatmap(human_fibrosis, width = unit(40, "mm"),name = "Fibrosis",
          column_order=paste0('F', seq(0,4)),
          col = circlize::colorRamp2(breaks=seq(-max(abs(human_fibrosis)), max(abs(human_fibrosis)), len
                                     colors=colPals$RdBu)) +
  Heatmap(mouse_log2FC_HFDmVsCDm, width = unit(8, "mm"), name = "Male mouse log2FC",
          col = colPals$RdBu[c(10,2)]) +
  Heatmap(mouse_log2FC_HFDfVsCDf, width = unit(8, "mm"), name = "Female mouse log2FC",
          col = c(colPals$RdBu[c(10,2)], '#F6F6F6'))
```



NAFLD classification models

NAFLD models

```
models <- list()
models[['Stage']] <- lapply(gene_sets, function(x) {
  buildModel(dat = t(cohort_data$Govaere$cpm_filt),
             group = factor(cohort_data$Govaere$meta$Stage, levels = c('CTRL','NAFL','NASH')),
             method = 'glmnet',
             features = x,
             preproc = c('scale', 'center'),
             train_test_split = 0.6,
             tune_iter = 10,
             seed = 22)
})
```

```
models[['NAS']] <- lapply(gene_sets, function(x) {
  buildModel(dat = t(cohort_data$Govaere$cpm_filt),
             group = factor(cohort_data$Govaere$meta$NAS_class, levels = c('low','mid','high')),
             method = 'glmnet',
             features = x,
             preproc = c('scale', 'center'),
             train_test_split = 0.6,
             tune_iter = 10,
             seed = 22)
})
models[['Fibrosis']] <- lapply(gene_sets, function(x) {
  buildModel(dat = t(cohort_data$Govaere$cpm_filt),
             group = factor(cohort_data$Govaere$meta$Fibrosis_class, levels = c('low','mid','high')),
             method = 'glmnet',
             features = x,
             preproc = c('scale', 'center'),
             train_test_split = 0.6,
             tune_iter = 10,
             seed = 22)
})

saveRDS(models, file = 'results/nafld_classification_models.rds')
```

Random gene set models

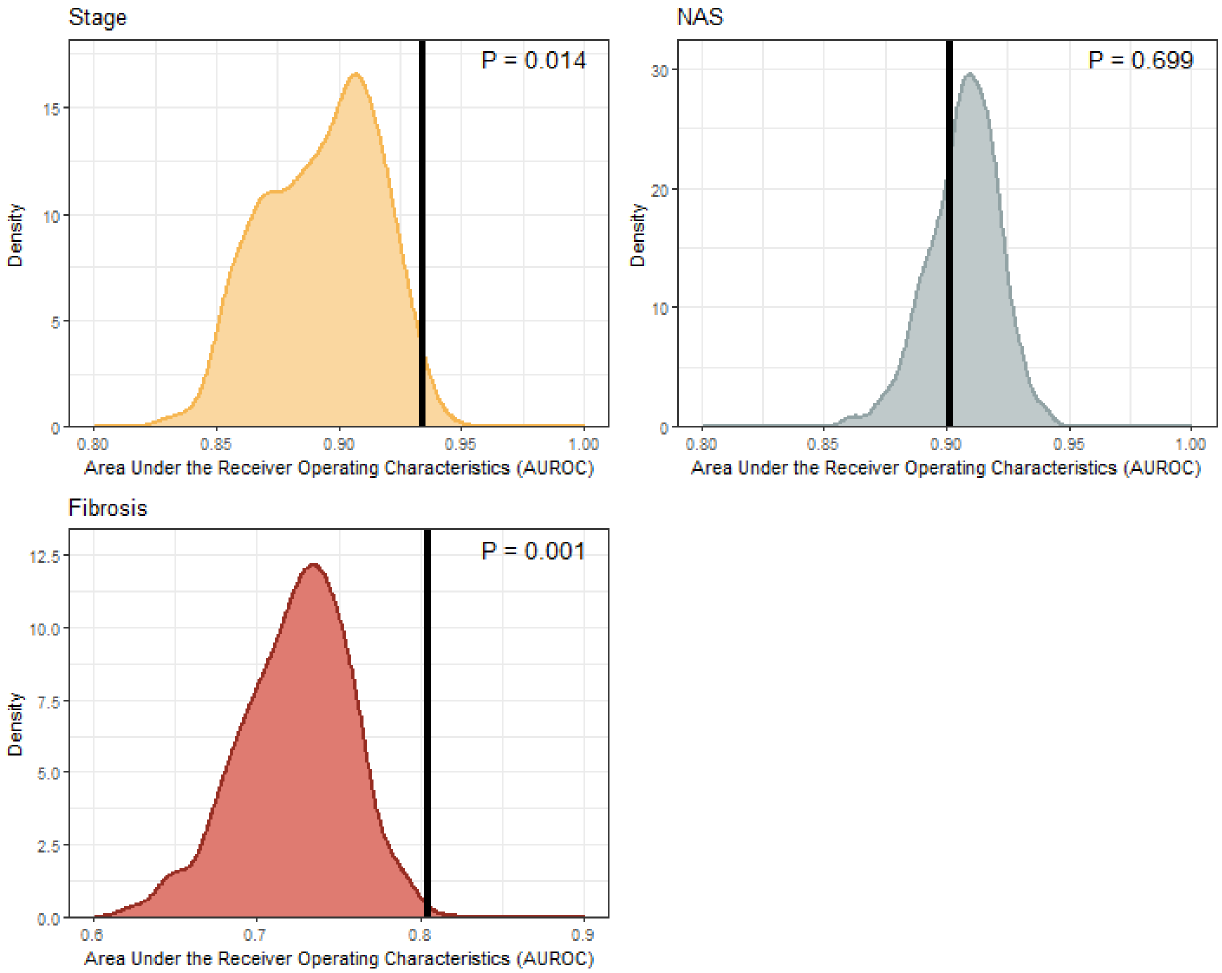
```
# random gene sets for null distribution of prediction models
# all_genes <- rownames(cohort_data$Govaere$cpm_filt)
# random_genes <- lapply(seq(1,1000), function(x) sample(x = all_genes, size = length(gene_sets$ER_gene
# saveRDS(random_genes, file = 'results/random_38gene_subsets.rds')

# load random gene sets for reproducibility
random_genes <- readRDS(file = 'results/random_38gene_subsets.rds')
```

```
# train random models (very long computing time!)
# random_models <- list()
# random_models[['Stage']] <- lapply(random_genes, function(x) {
#   buildModel(dat = t(cohort_data$Govaere$cpm_filt),
#              group = factor(cohort_data$Govaere$meta$Stage, levels = c('CTRL','NAFL','NASH')),
#              method = 'glmnet',
#              features = x,
#              preproc = c('scale', 'center'),
#              train_test_split = 0.6,
#              tune_iter = 10,
#              seed = 22)
# })
# random_models[['NAS']] <- lapply(random_genes, function(x) {
#   buildModel(dat = t(cohort_data$Govaere$cpm_filt),
#              group = factor(cohort_data$Govaere$meta$NAS_class, levels = c('low','mid','high')),
#              method = 'glmnet',
#              features = x,
#              preproc = c('scale', 'center'),
#              train_test_split = 0.6,
#              tune_iter = 10,
#               seed = 22)
# })
# random_models[['Fibrosis']] <- lapply(random_genes, function(x) {
#   buildModel(dat = t(cohort_data$Govaere$cpm_filt),
#               group = factor(cohort_data$Govaere$meta$Fibrosis_class, levels = c('low','mid','high')),
#               method = 'glmnet',
#               features = x,
#               preproc = c('scale', 'center'),
#               train_test_split = 0.6,
#               tune_iter = 10,
#               seed = 22)
# })

# saveRDS(random_models, file = 'Results/random_models_38genes.rds')

random_models <- readRDS(file = 'results/random_models_38genes.rds')
```

Predictive potential of ER regulated genes vs random gene sets

```
p <- list()

df <- data.frame(auc = lapply(random_models$Stage, function(x) x$resultsROC$AUC$model$micro) %>% unlist
val <- models$Stage$ER_genes$resultsROC$AUC$model$micro
pval <- tailFraction(val, df$auc, tail = 'right')
p[[1]] <- ggplot(df, aes(x = auc)) +
  geom_density(lwd = 1, colour = '#F6B651',
               fill = '#FAD7A0', alpha = 1) +
  geom_vline(xintercept = val, lwd = 1.5) +
  scale_x_continuous(limits = c(0.8,1)) +
  scale_y_continuous(expand = expansion(mult = c(0, .1))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('Area Under the Receiver Operating Characteristics (AUROC)') +
  ylab('Density') +
  ggtitle('Stage') +
  theme_bw()

df <- data.frame(auc = lapply(random_models$NAS, function(x) x$resultsROC$AUC$model$micro) %>% unlist()
val <- models$NAS$ER_genes$resultsROC$AUC$model$micro
pval <- tailFraction(val, df$auc, tail = 'right')
p[[2]] <- ggplot(df, aes(x = auc)) +
  geom_density(lwd = 1, colour = '#92A3A5',
               fill = '#BFC9CA', alpha = 1) +
  geom_vline(xintercept = val, lwd = 1.5) +
  scale_x_continuous(limits = c(0.8,1)) +
  scale_y_continuous(expand = expansion(mult = c(0, .1))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('Area Under the Receiver Operating Characteristics (AUROC)') +
  ylab('Density') +
  ggtitle('NAS') +
  theme_bw()

df <- data.frame(auc = lapply(random_models$Fibrosis, function(x) x$resultsROC$AUC$model$micro) %>% unl
val <- models$Fibrosis$ER_genes$resultsROC$AUC$model$micro
```

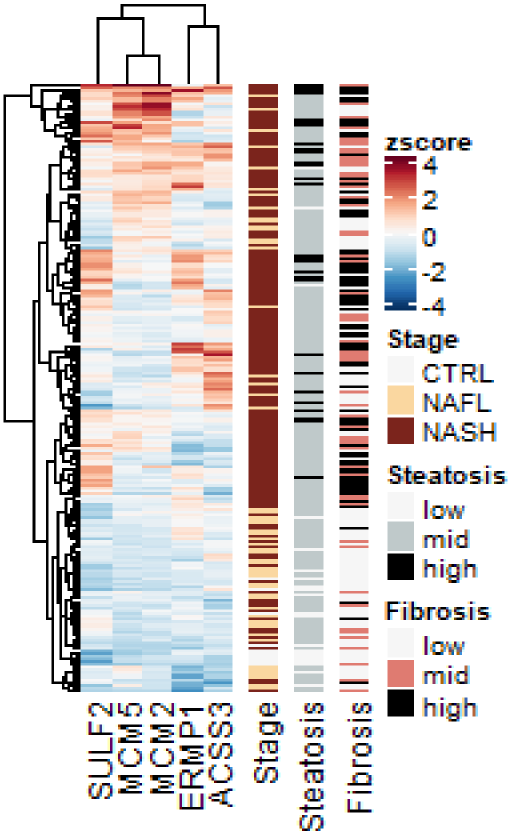
```
pval <- tailFraction(val, df$auc, tail = 'right')
p[[3]] <- ggplot(df, aes(x = auc)) +
  geom_density(lwd = 1, colour = '#972D22',
               fill = '#DF7B71', alpha = 1) +
  geom_vline(xintercept = val, lwd = 1.5, color = 'black') +
  scale_x_continuous(limits = c(0.6,0.9)) +
  scale_y_continuous(expand = expansion(mult = c(0, .1))) +
  annotate("text",  x=Inf, y = Inf, label = paste('P =',pval), vjust=1.5, hjust=1.2, size=5) +
  xlab('Area Under the Receiver Operating Characteristics (AUROC)') +
  ylab('Density') +
  ggtitle('Fibrosis') +
  theme_bw()

patchwork::wrap_plots(p, nrow=2, ncol=2, byrow=T)
```



NAS markers (patient stratification)

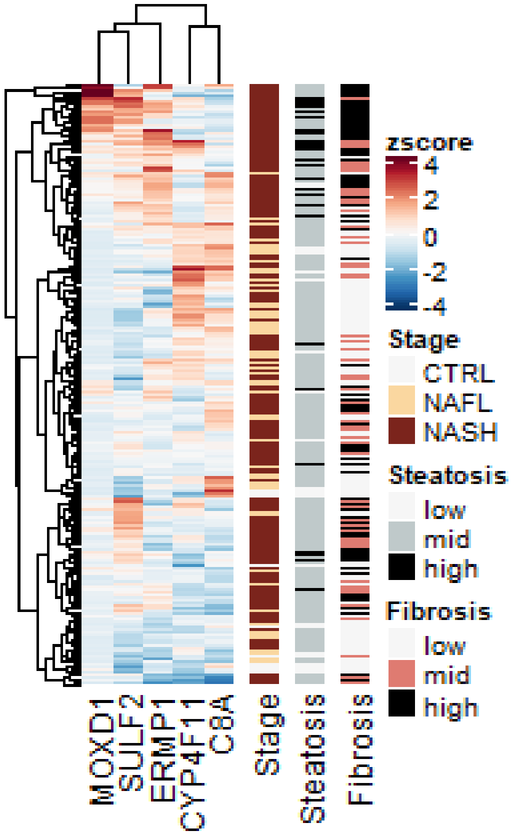
```
df <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% gene_sets$NAS_markers) %>%
  scaleData(method = 'zscore') %>%
  t()
```

```
Heatmap(df, width = unit(20, "mm"),name = "zscore", cluster_columns = T,
        col = circlize::colorRamp2(breaks=seq(-4, 4, length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(cohort_data$Govaere$meta$Stage %>% factor(levels = c('CTRL','NAFL','NASH')),
          width = unit(4, "mm"), name = "Stage",
          col = colPals$stage) +
  Heatmap(cohort_data$Govaere$meta$NAS_class %>% factor(levels = c('low','mid','high')),
          width = unit(4, "mm"), name = "Steatosis",
          col = colPals$NAS) +
  Heatmap(cohort_data$Govaere$meta$Fibrosis_class %>% factor(levels = c('low','mid','high')),
          width = unit(4, "mm"), name = "Fibrosis",
          col = colPals$fibrosis)
```



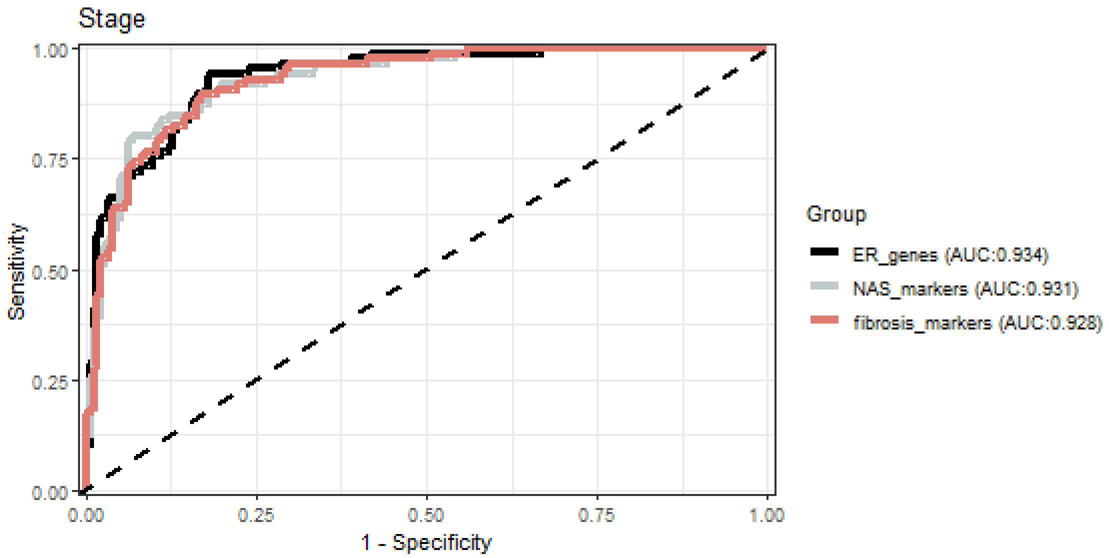
Fibrosis markers (patient stratification)

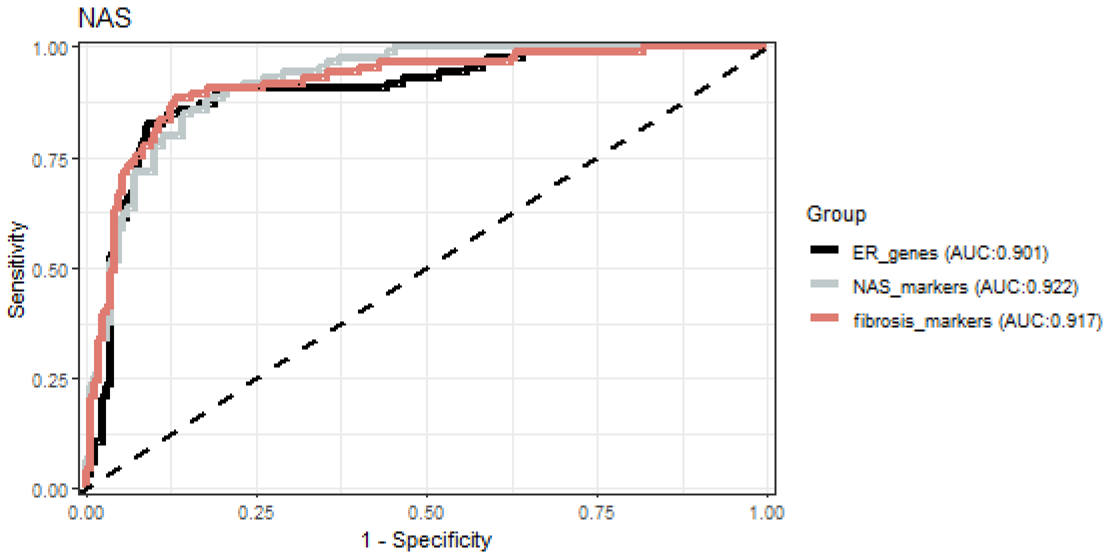
```
df <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% gene_sets$fibrosis_markers) %>%
  scaleData(method = 'zscore') %>%
  t()

Heatmap(df, width = unit(20, "mm"),name = "zscore", cluster_columns = T,
        col = circlize::colorRamp2(breaks=seq(-4, 4, length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(cohort_data$Govaere$meta$Stage %>% factor(levels = c('CTRL','NAFL','NASH')),
          width = unit(4, "mm"), name = "Stage",
          col = colPals$stage) +
  Heatmap(cohort_data$Govaere$meta$NAS_class %>% factor(levels = c('low','mid','high')),
          width = unit(4, "mm"), name = "Steatosis",
          col = colPals$NAS) +
  Heatmap(cohort_data$Govaere$meta$Fibrosis_class %>% factor(levels = c('low','mid','high')),
          width = unit(4, "mm"), name = "Fibrosis",
          col = colPals$fibrosis)
```

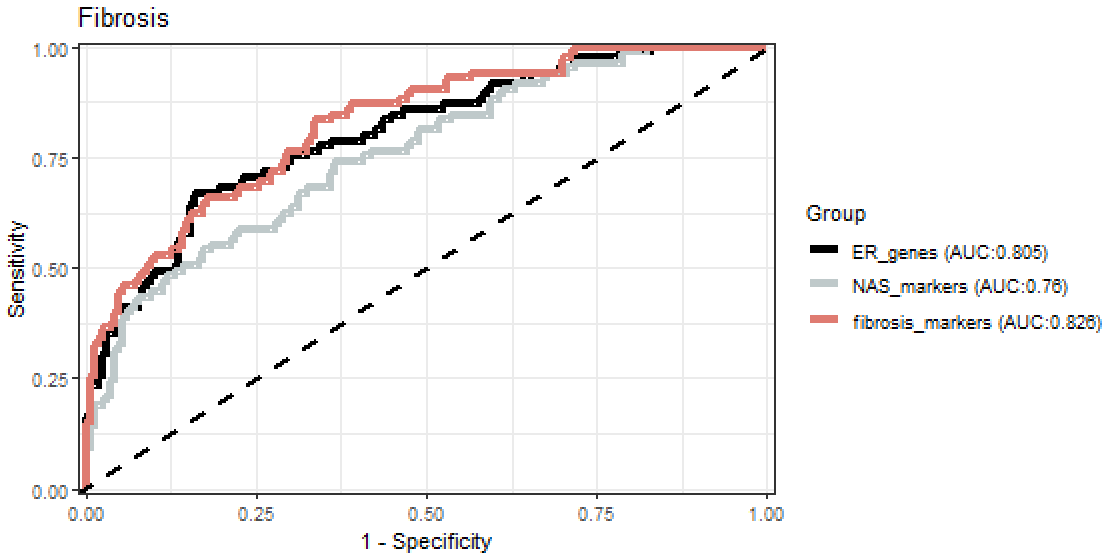


ROC curves

```
p <- list()

df <- lapply(models$Stage[1:3], function(x) x$plotROC %>% dplyr::filter(Group == 'Micro'))
df <- lapply(names(models$Stage[1:3]), function(x) df[[x]] %>% dplyr::mutate(Group = x))
df <- dplyr::bind_rows(df) %>%
  dplyr::mutate(Group = factor(Group, levels = c('ER_genes','NAS_markers','fibrosis_markers')),
                label = paste0(Group, ' (AUC:', round(AUC,3), ')'))

p[[1]] <- ggplot(df, aes(x=1-Specificity, y=Sensitivity)) +
  geom_path(aes(color=Group), size=1.5) +
  geom_abline(intercept=0, slope=1, color='#000000', lwd=1, linetype = 'dashed') +
  scale_x_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_color_manual(values = c('#000000','#BFC9CA','#DF7B71'), labels=unique(df$label)) +
  ggtitle('Stage') +
  theme_bw()

df <- lapply(models$NAS[1:3], function(x) x$plotROC %>% dplyr::filter(Group == 'Micro'))
df <- lapply(names(models$NAS[1:3]), function(x) df[[x]] %>% dplyr::mutate(Group = x))
df <- dplyr::bind_rows(df) %>%
  dplyr::mutate(Group = factor(Group, levels = c('ER_genes','NAS_markers','fibrosis_markers')),
                label = paste0(Group, ' (AUC:', round(AUC,3), ')'))

p[[2]] <- ggplot(df, aes(x=1-Specificity, y=Sensitivity)) +
  geom_path(aes(color=Group), size=1.5) +
  geom_abline(intercept=0, slope=1, color='#000000', lwd=1, linetype = 'dashed') +
  scale_x_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_color_manual(values = c('#000000','#BFC9CA','#DF7B71'), labels=unique(df$label)) +
  ggtitle('NAS') +
  theme_bw()

df <- lapply(models$Fibrosis[1:3], function(x) x$plotROC %>% dplyr::filter(Group == 'Micro'))
df <- lapply(names(models$Fibrosis[1:3]), function(x) df[[x]] %>% dplyr::mutate(Group = x))
df <- dplyr::bind_rows(df) %>%
  dplyr::mutate(Group = factor(Group, levels = c('ER_genes','NAS_markers','fibrosis_markers')),
                label = paste0(Group, ' (AUC:', round(AUC,3), ')'))

p[[3]] <- ggplot(df, aes(x=1-Specificity, y=Sensitivity)) +
  geom_path(aes(color=Group), size=1.5) +
  geom_abline(intercept=0, slope=1, color='#000000', lwd=1, linetype = 'dashed') +
  scale_x_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_color_manual(values = c('#000000','#BFC9CA','#DF7B71'), labels=unique(df$label)) +
  ggtitle('Fibrosis') +
  theme_bw()

patchwork::wrap_plots(p, nrow=3, ncol=1, byrow=T)
```

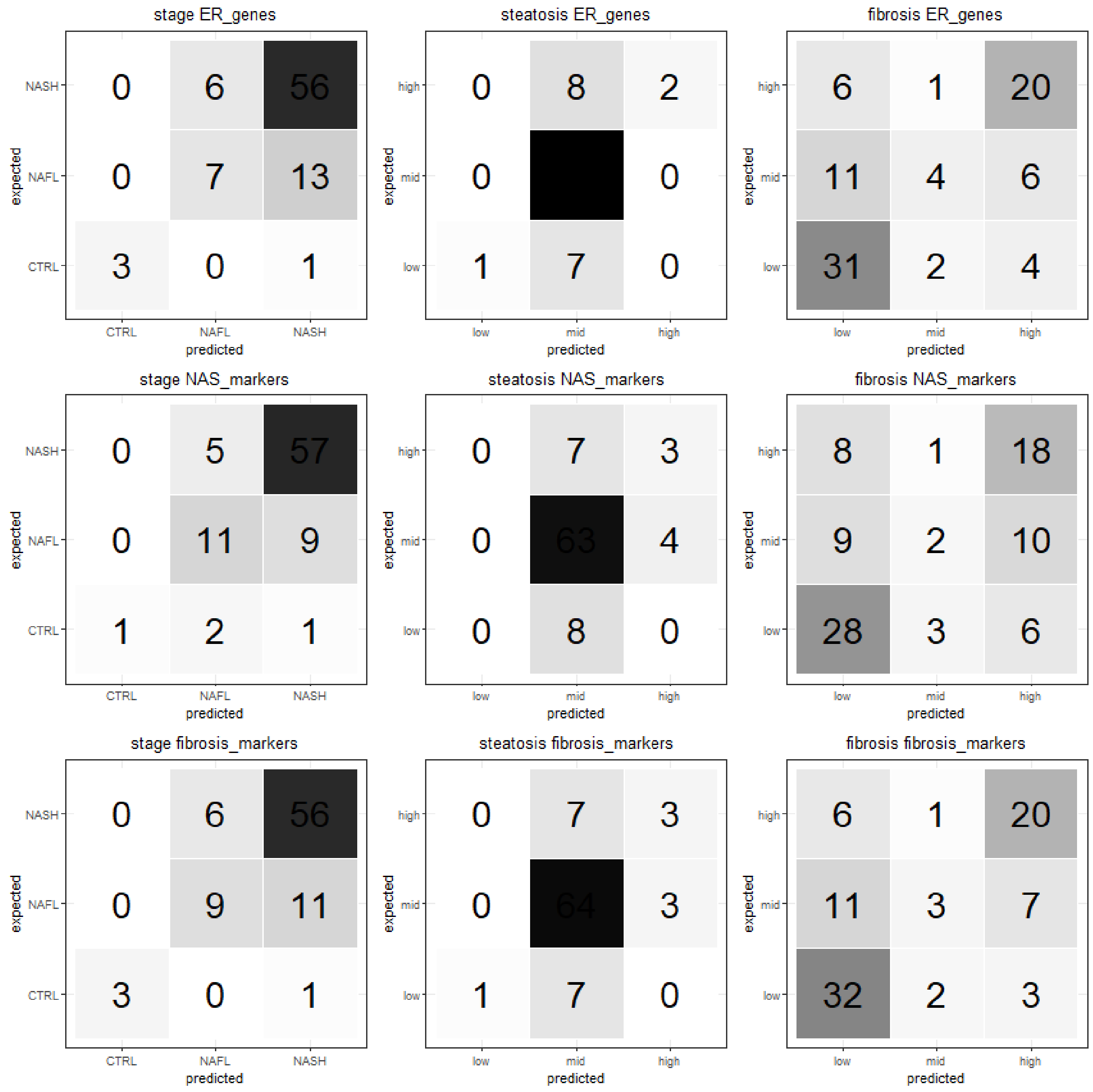

Stage
Sensitivity
1 - Specificity
0.00
0.25
0.50
0.75
1.00
0.00
0.25
0.50
0.75
1.00
Group
ER_genes (AUC:0.934)
NAS_markers (AUC:0.931)
fibrosis_markers (AUC:0.928)


NAS
Sensitivity
1 - Specificity
0.00
0.25
0.50
0.75
1.00
0.00
0.25
0.50
0.75
1.00
Group
ER_genes (AUC:0.901)
NAS_markers (AUC:0.922)
fibrosis_markers (AUC:0.917)


Fibrosis
Sensitivity
1 - Specificity
0.00
0.25
0.50
0.75
1.00
0.00
0.25
0.50
0.75
1.00
Group
ER_genes (AUC:0.805)
NAS_markers (AUC:0.76)
fibrosis_markers (AUC:0.826)

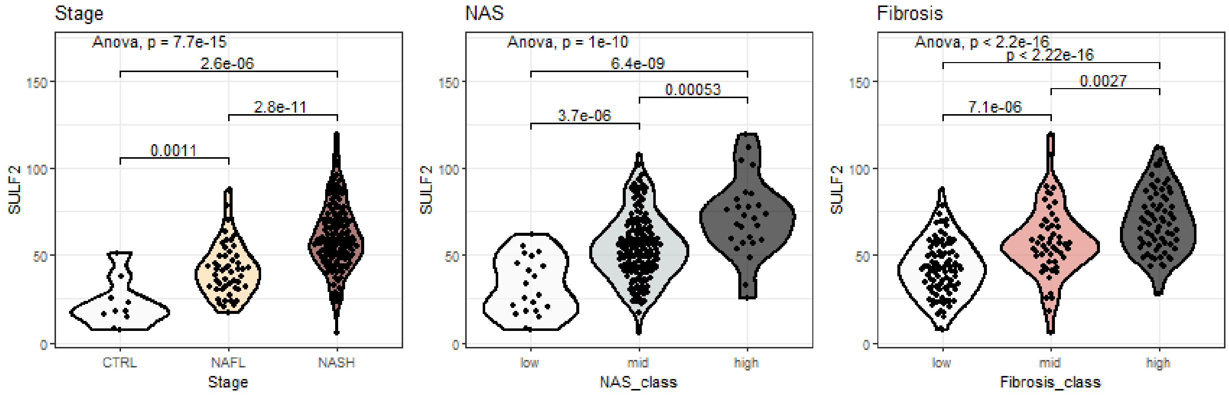
Confusion matrices

```r
df <- lapply(names(models$Stage[1:3]), function(x) models$Stage[[x]]$confusionMat %>% dplyr::mutate(con
df <- c(df, lapply(names(models$NAS[1:3]), function(x) models$NAS[[x]]$confusionMat %>% dplyr::mutate(c
df <- c(df, lapply(names(models$Fibrosis[1:3]), function(x) models$Fibrosis[[x]]$confusionMat %>% dplyr
max_count <- lapply(df, function(x) x %>% dplyr::pull(count)) %>% unlist() %>% max()
df <- lapply(df, function(x) x %>% dplyr::mutate(color=paste0('grey', 100-round(count/max_count*100))))
p <- lapply(df, function(x) {
  ggplot(x, aes(predicted,expected)) +
    geom_tile(aes(fill = color), color = "white") +
    geom_text(aes(label = sprintf("%1.0f", count)), vjust = 0.5, size=10) +
    scale_fill_identity() +
    theme_void() +
    theme_bw() +
    ggtitle(paste(x$cond[1], x$set[1])) +
    theme(legend.position = "none",
          plot.title = element_text(hjust = 0.5))
})

patchwork::wrap_plots(p, nrow=3, ncol=3, byrow=F)
```



SULF2 expression across NAFLD stage, NAS and fibrosis categories

```
p <- list()

df <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% c('SULF2', 'THBS2')) %>%
  t() %>%
  cbind(cohort_data$Govaere$meta) %>%
  dplyr::mutate(Stage=factor(Stage, levels=c('CTRL', 'NAFL', 'NASH')),
                NAS_class=factor(NAS_class, levels=c('low', 'mid', 'high')),
                Fibrosis_class=factor(Fibrosis_class, levels=c('low', 'mid', 'high')))

p[[1]] <- ggplot(df, aes(x=Stage, y=SULF2, fill=Stage, color=Stage)) +
  geom_violin(lwd=1, alpha=0.6, color='black') +
  ggbeeswarm::geom_quasirandom(width = 0.2, color='black') +
  stat_compare_means(method = 't.test',
                     comparisons = list(c('CTRL','NAFL'), c('NAFL','NASH'), c('CTRL','NASH')),
                     label.y = c(100,125,150)) +
  stat_compare_means(method = 'anova', label.y = 170) +
  scale_fill_manual(values = colPals$stage) +
  scale_color_manual(values = colPals$stage) +
  ggtitle('Stage') +
  theme_bw() +
  theme(legend.position='none')

p[[2]] <- ggplot(df, aes(x=NAS_class, y=SULF2, fill=NAS_class, color=NAS_class)) +
  geom_violin(lwd=1, alpha=0.6, color='black') +
  ggbeeswarm::geom_quasirandom(width = 0.2, color='black') +
  stat_compare_means(method = 't.test',
                     comparisons = list(c('low','mid'), c('mid','high'), c('low','high')),
                     label.y = c(120,135,150)) +
  stat_compare_means(method = 'anova', label.y = 170) +
  scale_fill_manual(values = colPals$NAS) +
  scale_color_manual(values = colPals$NAS) +
  ggtitle('NAS') +
  theme_bw() +
  theme(legend.position='none')

p[[3]] <- ggplot(df, aes(x=Fibrosis_class, y=SULF2, fill=Fibrosis_class, color=Fibrosis_class)) +
  geom_violin(lwd=1, alpha=0.6, color='black') +
  ggbeeswarm::geom_quasirandom(width = 0.2, color='black') +
  stat_compare_means(method = 't.test',
                     comparisons = list(c('low','mid'), c('mid','high'), c('low','high')),
                     label.y = c(125,140,155)) +
  stat_compare_means(method = 'anova', label.y = 170) +
  scale_fill_manual(values = colPals$fibrosis) +
  scale_color_manual(values = colPals$fibrosis) +
  ggtitle('Fibrosis') +
  theme_bw() +
  theme(legend.position='none')

patchwork::wrap_plots(p, nrow=1, ncol=3, byrow=T)
```



```
# significance of multiple comparisons from ANOVA using Tukey
SULF2_Stage_aov <- aov(df$SULF2 ~ df$Stage)
SULF2_fib_aov <- aov(df$SULF2 ~ df$Fibrosis_class)
SULF2_NAS_aov <- aov(df$SULF2 ~ df$NAS_class)
TukeyHSD(SULF2_Stage_aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$SULF2 ~ df$Stage)
##
## $`df$Stage`
##                diff       lwr      upr    p adj
## NAFL-CTRL 19.59177  4.689907 34.49364 0.006137
## NASH-CTRL 38.86036 24.801929 52.91880 0.000000
## NASH-NAFL 19.26859 12.312847 26.22433 0.000000
```

```
TukeyHSD(SULF2_fib_aov)
```

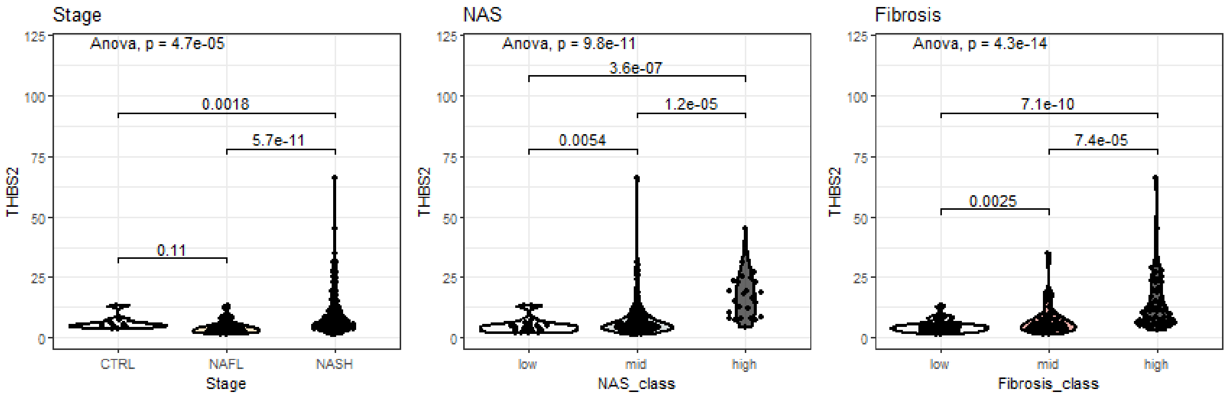
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$SULF2 ~ df$Fibrosis_class)
##
## $`df$Fibrosis_class`
##               diff       lwr      upr     p adj
## mid-low  15.47130  8.266206 22.67639 0.0000026
## high-low 26.37990 19.662390 33.09742 0.0000000
## high-mid 10.90861  3.217335 18.59988 0.0027620
```

```
TukeyHSD(SULF2_NAS_aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$SULF2 ~ df$NAS_class)
##
## $`df$NAS_class`
##               diff       lwr      upr    p adj
## mid-low  22.57695 12.153597 33.00030 2.10e-06
## high-low 40.14145 26.924343 53.35855 0.00e+00
## high-mid 17.56450  8.074392 27.05460 5.79e-05
```

THBS2 expression across NAFLD stage, NAS and fibrosis categories

```
p <- list()

p[[1]] <- ggplot(df, aes(x=Stage, y=THBS2, fill=Stage, color=Stage)) +
  geom_violin(lwd=1, alpha=0.6, color='black') +
  ggbeeswarm::geom_quasirandom(width = 0.2, color='black') +
  stat_compare_means(method = 't.test',
                     comparisons = list(c('CTRL','NAFL'), c('NAFL','NASH'), c('CTRL','NASH')),
                     label.y = c(30,75,90)) +
  stat_compare_means(method = 'anova', label.y = 120) +
  scale_fill_manual(values = colPals$stage) +
  scale_color_manual(values = colPals$stage) +
  ggtitle('Stage') +
  theme_bw() +
  theme(legend.position='none')

p[[2]] <- ggplot(df, aes(x=NAS_class, y=THBS2, fill=NAS_class, color=NAS_class)) +
  geom_violin(lwd=1, alpha=0.6, color='black') +
  ggbeeswarm::geom_quasirandom(width = 0.2, color='black') +
  stat_compare_means(method = 't.test',
                     comparisons = list(c('low','mid'), c('mid','high'), c('low','high')),
                     label.y = c(75,90,105)) +
  stat_compare_means(method = 'anova', label.y = 120) +
  scale_fill_manual(values = colPals$NAS) +
  scale_color_manual(values = colPals$NAS) +
  ggtitle('NAS') +
  theme_bw() +
  theme(legend.position='none')

p[[3]] <- ggplot(df, aes(x=Fibrosis_class, y=THBS2, fill=Fibrosis_class, color=Fibrosis_class)) +
  geom_violin(lwd=1, alpha=0.6, color='black') +
  ggbeeswarm::geom_quasirandom(width = 0.2, color='black') +
  stat_compare_means(method = 't.test',
                     comparisons = list(c('low','mid'), c('mid','high'), c('low','high')),
                     label.y = c(50,75,90)) +
  stat_compare_means(method = 'anova', label.y = 120) +
  scale_fill_manual(values = colPals$fibrosis) +
  scale_color_manual(values = colPals$fibrosis) +
  ggtitle('Fibrosis') +
  theme_bw() +
  theme(legend.position='none')

patchwork::wrap_plots(p, nrow=1, ncol=3, byrow=T)
```



```
# significance of multiple comparisons from ANOVA using Tukey
THBS2_Stage_aov <- aov(df$THBS2 ~ df$Stage)
THBS2_fib_aov <- aov(df$THBS2 ~ df$Fibrosis_class)
THBS2_NAS_aov <- aov(df$THBS2 ~ df$NAS_class)
TukeyHSD(THBS2_Stage_aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$THBS2 ~ df$Stage)
##
## $`df$Stage`
##                diff       lwr       upr     p adj
## NAFL-CTRL -1.648736 -8.070500  4.773028 0.8170167
## NASH-CTRL  4.009379 -2.048919 10.067677 0.2644515
## NASH-NAFL  5.658114  2.660628  8.655601 0.0000402
```

```
TukeyHSD(THBS2_fib_aov)
```

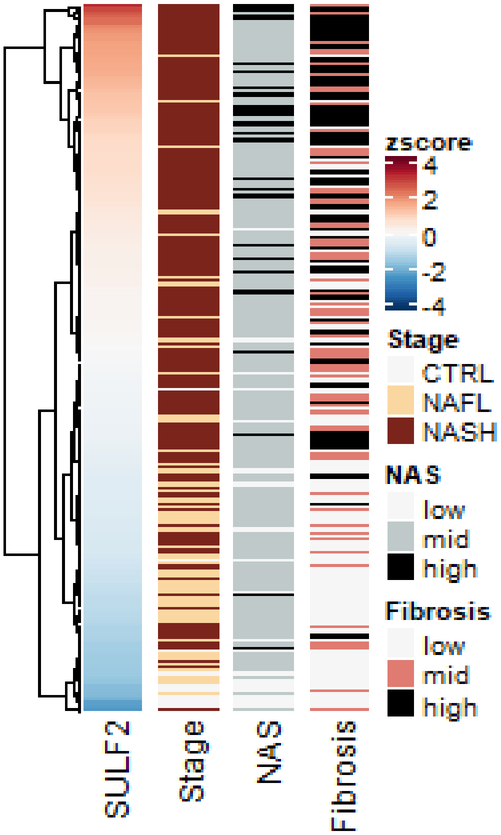
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$THBS2 ~ df$Fibrosis_class)
##
## $`df$Fibrosis_class`
##              diff        lwr       upr     p adj
## mid-low  2.963397 0.08796513  5.838830 0.0417025
## high-low 9.538727 6.85787890 12.219575 0.0000000
## high-mid 6.575330 3.50587183  9.644787 0.0000028
```

```
TukeyHSD(THBS2_NAS_aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = df$THBS2 ~ df$NAS_class)
##
## $`df$NAS_class`
##               diff       lwr       upr     p adj
## mid-low   2.570937 -1.469549  6.611424 0.2921252
## high-low 13.138777  8.015328 18.262227 0.0000000
## high-mid 10.567840  6.889117 14.246563 0.0000000
```

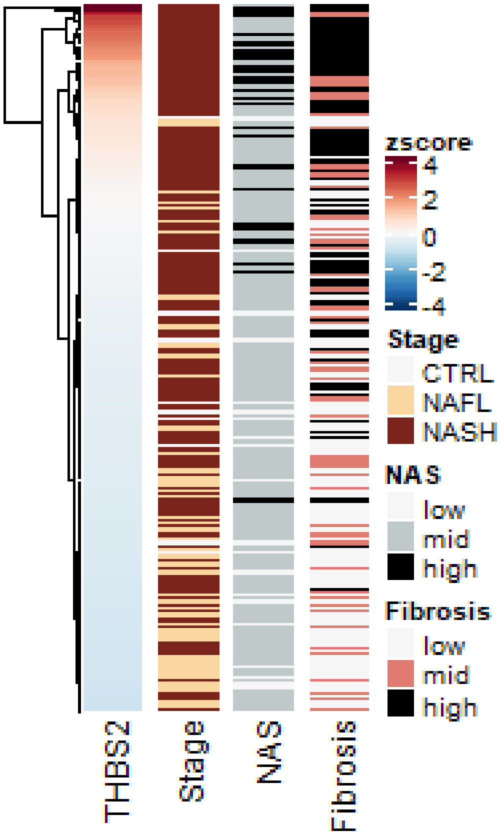
SULF2 expression (patient stratification)

```
df <- cohort_data$Govaere$cpm_filt %>%
  dplyr::filter(rownames(.) %in% c('SULF2', 'THBS2')) %>%
  scaleData(method = 'zscore') %>%
  t() %>%
  cbind(cohort_data$Govaere$meta) %>%
  dplyr::mutate(Stage=factor(Stage, levels=c('CTRL', 'NAFL', 'NASH')),
                NAS_class=factor(NAS_class, levels=c('low', 'mid', 'high')),
                Fibrosis_class=factor(Fibrosis_class, levels=c('low', 'mid', 'high')))

Heatmap(select(df, SULF2), width = unit(8, "mm"),name = "zscore", cluster_columns = T,
        col = circlize::colorRamp2(breaks=seq(-4, 4, length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(df$Stage, width = unit(8, "mm"), name = "Stage",col = colPals$stage) +
  Heatmap(df$NAS_class, width = unit(8, "mm"), name = "NAS",col = colPals$NAS) +
  Heatmap(df$Fibrosis_class, width = unit(8, "mm"), name = "Fibrosis",col = colPals$fibrosis)
```



THBS2 expression (patient stratification)

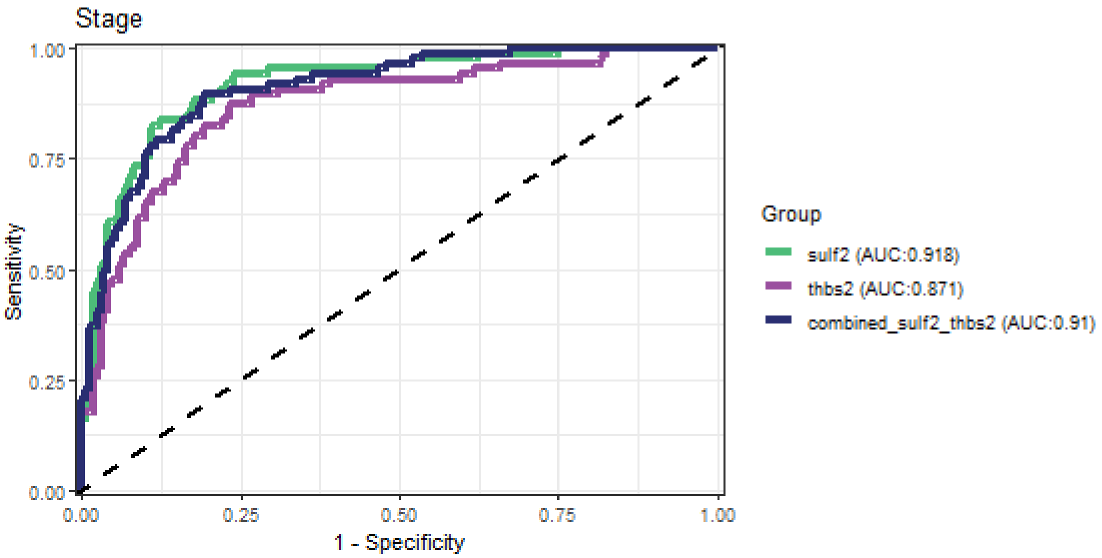
```
Heatmap(select(df, THBS2), width = unit(8, "mm"),name = "zscore", cluster_columns = T,
        col = circlize::colorRamp2(breaks=seq(-4, 4, length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(df$Stage, width = unit(8, "mm"), name = "Stage",col = colPals$stage) +
  Heatmap(df$NAS_class, width = unit(8, "mm"), name = "NAS",col = colPals$NAS) +
  Heatmap(df$Fibrosis_class, width = unit(8, "mm"), name = "Fibrosis",col = colPals$fibrosis)
```

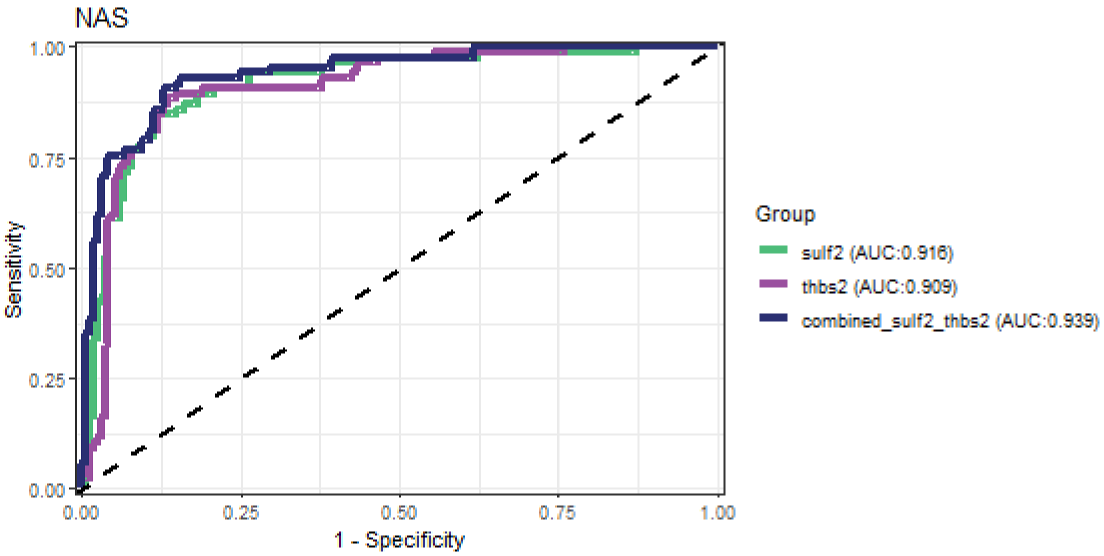


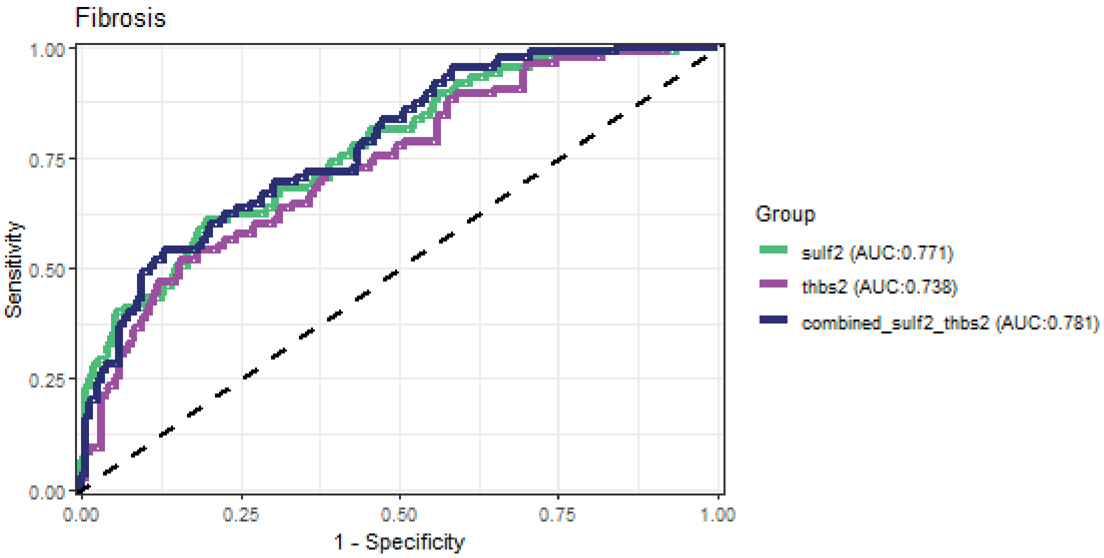
ROC curves of SULF2 & THBS2 classification models

```
p <- list()

df <- lapply(models$Stage[4:6], function(x) x$plotROC %>% dplyr::filter(Group == 'Micro'))
df <- lapply(names(models$Stage[4:6]), function(x) df[[x]] %>% dplyr::mutate(Group = x))
df <- dplyr::bind_rows(df) %>%
  dplyr::mutate(Group = factor(Group, levels = c('sulf2','thbs2','combined_sulf2_thbs2')),
                label = paste0(Group, ' (AUC:', round(AUC,3), ')'))

p[[1]] <- ggplot(df, aes(x=1-Specificity, y=Sensitivity)) +
  geom_path(aes(color=Group), size=1.5) +
  geom_abline(intercept=0, slope=1, color='#000000', lwd=1, linetype = 'dashed') +
  scale_x_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_color_manual(values = c('#4DBC79','#9A509F','#2A2F72'), labels=unique(df$label)) +
  ggtitle('Stage') +
  theme_bw()

df <- lapply(models$NAS[4:6], function(x) x$plotROC %>% dplyr::filter(Group == 'Micro'))
df <- lapply(names(models$NAS[4:6]), function(x) df[[x]] %>% dplyr::mutate(Group = x))
df <- dplyr::bind_rows(df) %>%
  dplyr::mutate(Group = factor(Group, levels = c('sulf2','thbs2','combined_sulf2_thbs2')),
                label = paste0(Group, ' (AUC:', round(AUC,3), ')'))
```

```r
p[[2]] <- ggplot(df, aes(x=1-Specificity, y=Sensitivity)) +
  geom_path(aes(color=Group), size=1.5) +
  geom_abline(intercept=0, slope=1, color='#000000', lwd=1, linetype = 'dashed') +
  scale_x_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_color_manual(values = c('#4DBC79','#9A509F','#2A2F72'), labels=unique(df$label)) +
  ggtitle('NAS') +
  theme_bw()

df <- lapply(models$Fibrosis[4:6], function(x) x$plotROC %>% dplyr::filter(Group == 'Micro'))
df <- lapply(names(models$Fibrosis[4:6]), function(x) df[[x]] %>% dplyr::mutate(Group = x))
df <- dplyr::bind_rows(df) %>%
  dplyr::mutate(Group = factor(Group, levels = c('sulf2','thbs2','combined_sulf2_thbs2')),
                label = paste0(Group, ' (AUC:', round(AUC,3), ')'))

p[[3]] <- ggplot(df, aes(x=1-Specificity, y=Sensitivity)) +
  geom_path(aes(color=Group), size=1.5) +
  geom_abline(intercept=0, slope=1, color='#000000', lwd=1, linetype = 'dashed') +
  scale_x_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_color_manual(values = c('#4DBC79','#9A509F','#2A2F72'), labels=unique(df$label)) +
  ggtitle('Fibrosis') +
  theme_bw()

patchwork::wrap_plots(p, nrow=3, ncol=1, byrow=T)
```


Stage
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 - Specificity
Group
sulf2 (AUC:0.918)
thbs2 (AUC:0.871)
combined_sulf2_thbs2 (AUC:0.91)


NAS
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 - Specificity
Group
sulf2 (AUC:0.916)
thbs2 (AUC:0.909)
combined_sulf2_thbs2 (AUC:0.939)


Fibrosis
Sensitivity
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1 - Specificity
Group
sulf2 (AUC:0.771)
thbs2 (AUC:0.738)
combined_sulf2_thbs2 (AUC:0.781)

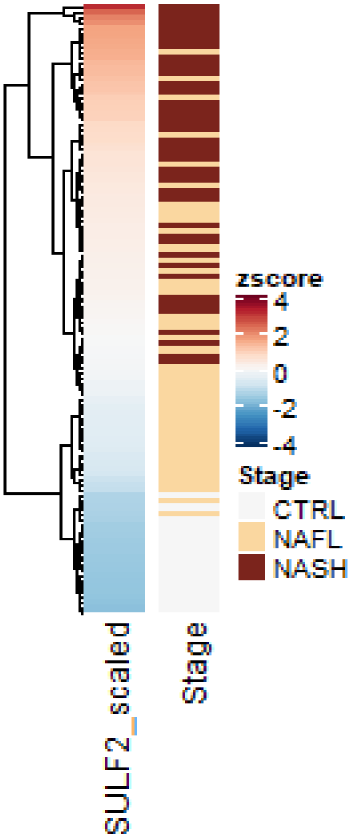
External validation

Integrate GTEx v8 (CTRL) & Kozumi et al. 2021 (NAFL/NASH) datasets

```
# merge cohort data from GTEx v8 (CTRL) & Kozumi et al. 2021 (NAFL/NASH)
eval_dat <- cohort_data$GTEx_liver_CTRL$cpm_filt %>%
  tibble::rownames_to_column(var = 'GeneSymbol') %>%
  dplyr::inner_join(tibble::rownames_to_column(cohort_data$Kozumi$cpm_filt, var = 'GeneSymbol'), by='Gen
  tibble::column_to_rownames(var = 'GeneSymbol')

eval_dat_meta <- data.frame(Sample=c(colnames(cohort_data$Kozumi$cpm_filt),
                                 colnames(cohort_data$GTEx_liver_CTRL$cpm_filt)),
                        Stage=c(cohort_data$Kozumi$meta$Stage,
                                rep('CTRL', dim(cohort_data$GTEx_liver_CTRL$meta)[1])),
                        Source=c(rep('Kozumi', dim(cohort_data$Kozumi$meta)[1]),
                                 rep('GTEx', dim(cohort_data$GTEx_liver_CTRL$meta)[1]))) %>%
  dplyr::arrange(factor(Sample, levels = colnames(eval_dat)))

# normalize expression across datasets using a stable houskeeping gene
# SRSF4 has been reported as stably expressed throughout NAFLD spectrum
# (https://doi.org/10.1111/j.1530-0277.2011.01627.x)
df <- eval_dat %>%
  dplyr::filter(rownames(.) %in% c('SULF2', 'THBS2', 'SRSF4')) %>%
  t() %>%
  as.data.frame() %>%
  dplyr::mutate(SULF2_scaled=SULF2/(SRSF4/mean(SRSF4)),
                THBS2_scaled=THBS2/(SRSF4/mean(SRSF4)),
                SRSF4_scaled=SRSF4/(SRSF4/mean(SRSF4))) %>%
  t() %>%
  scaleData(method = 'zscore') %>%
  t() %>%
  cbind(eval_dat_meta) %>%
  dplyr::mutate(Stage=factor(Stage, levels=c('CTRL', 'NAFL', 'NASH')),
                Source=factor(Source, levels=c('Kozumi', 'GTEx')))
```

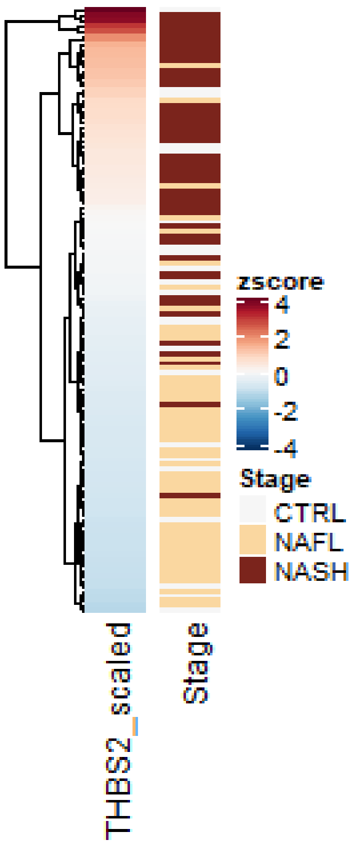
SULF2 expression (patient stratification)

```
Heatmap(select(df, SULF2_scaled), width = unit(8, "mm"),name = "zscore", cluster_columns = T,
        col = circlize::colorRamp2(breaks=seq(-4, 4, length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(df$Stage, width = unit(8, "mm"), name = "Stage", col = colPals$stage)
```



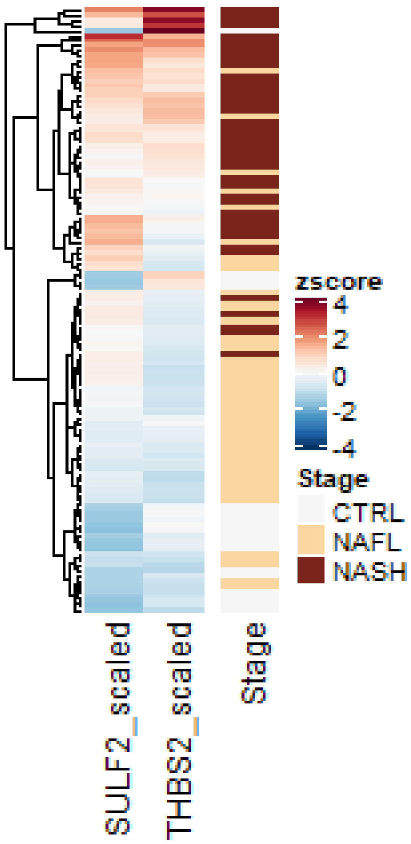
THBS2 expression (patient stratification)

```
Heatmap(select(df, THBS2_scaled), width = unit(8, "mm"),name = "zscore", cluster_columns = T,
        col = circlize::colorRamp2(breaks=seq(-4, 4, length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(df$Stage, width = unit(8, "mm"), name = "Stage", col = colPals$stage)
```



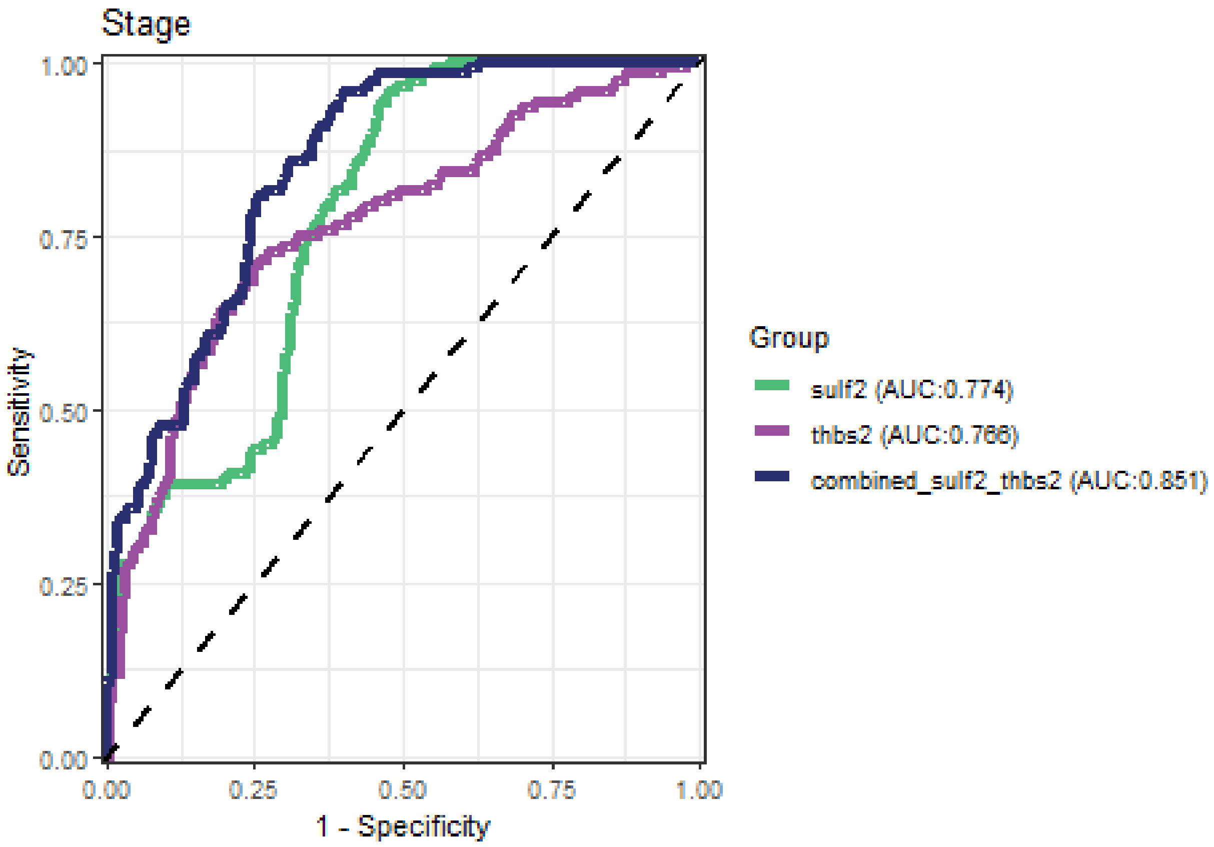
Combined SULF2 & THBS2 expression (patient stratification)

```
Heatmap(select(df, SULF2_scaled, THBS2_scaled), width = unit(16, "mm"),name = "zscore", cluster_columns
        col = circlize::colorRamp2(breaks=seq(-4, 4, length.out=11),
                                   colors=colPals$RdBu)) +
  Heatmap(df$Stage, width = unit(8, "mm"), name = "Stage", col = colPals$stage)
```



ROC curves for SULF2 & THBS2 model predictions

```
scaling_factors <- eval_dat %>%
  t() %>%
  as.data.frame() %>%
  dplyr::mutate(SRSF4=SRSF4/mean(SRSF4)) %>%
  dplyr::pull(SRSF4)

eval_dat_scaled <- t(t(eval_dat)/scaling_factors)

tests <- list()
tests[['Stage']][['sulf2']] <- testModel(models$Stage$sulf2$model,
                                          t(eval_dat_scaled),
                                          eval_dat_meta$Stage,
                                          preproc = c('scale', 'center'))
tests[['Stage']][['thbs2']] <- testModel(models$Stage$thbs2$model,
                                          t(eval_dat_scaled),
                                          eval_dat_meta$Stage,
                                          preproc = c('scale', 'center'))
tests[['Stage']][['combined_sulf2_thbs2']] <- testModel(models$Stage$combined_sulf2_thbs2$model,
                                                        t(eval_dat_scaled),
                                                        eval_dat_meta$Stage,
                                                        preproc = c('scale', 'center'))

df <- lapply(tests$Stage, function(x) x$plotROC %>% dplyr::filter(Group == 'Micro'))
df <- lapply(names(tests$Stage), function(x) df[[x]] %>% dplyr::mutate(Group = x))
df <- dplyr::bind_rows(df) %>%
  dplyr::mutate(Group = factor(Group, levels = c('sulf2','thbs2','combined_sulf2_thbs2')),
                label = paste0(Group, ' (AUC:', round(AUC,3), ')'))

ggplot(df, aes(x =1-Specificity, y=Sensitivity)) +
  geom_path(aes(color=Group), size=1.5) +
  geom_abline(intercept=0, slope=1, color='#000000', lwd=1, linetype = 'dashed') +
  scale_x_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_y_continuous(expand = expansion(mult = c(.01, .01))) +
  scale_color_manual(values = c('#4DBC79','#9A509F','#2A2F72'), labels=unique(df$label)) +
  ggtitle('Stage') +
  theme_bw()
```



```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ggpubr_0.4.0         patchwork_1.1.1      multiROC_1.1.1
##  [4] caret_6.0-92         lattice_0.20-41      ComplexHeatmap_2.4.3
##  [7] RColorBrewer_1.1-2   forcats_0.5.1        stringr_1.4.0
## [10] dplyr_1.0.3          purrr_0.3.4          readr_1.4.0
## [13] tidyr_1.2.0          tibble_3.1.4         ggplot2_3.3.3
## [16] tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##   [1] ggbeeswarm_0.6.0     colorspace_2.0-0     ggsignif_0.6.3
##   [4] rjson_0.2.20         ellipsis_0.3.2       class_7.3-18
##   [7] circlize_0.4.12      GlobalOptions_0.1.2  fs_1.5.0
##  [10] clue_0.3-58          rstudioapi_0.13      farver_2.0.3
##  [13] listenv_0.8.0        prodlim_2019.11.13   fansi_0.4.2
##  [16] lubridate_1.7.9.2    xml2_1.3.2           codetools_0.2-18
##  [19] splines_4.0.5        knitr_1.31           jsonlite_1.7.2
##  [22] pROC_1.18.0          broom_0.7.4          cluster_2.1.1
##  [25] dbplyr_2.0.0         png_0.1-7            compiler_4.0.5
##  [28] httr_1.4.2           backports_1.2.1      assertthat_0.2.1
##  [31] Matrix_1.3-2         fastmap_1.1.0        cli_2.3.0
##  [34] htmltools_0.5.2      tools_4.0.5          gtable_0.3.0
##  [37] glue_1.4.2           reshape2_1.4.4       Rcpp_1.0.7
##  [40] carData_3.0-5        cellranger_1.1.0     vctrs_0.3.8
##  [43] nlme_3.1-151         iterators_1.0.13     timeDate_3043.102
##  [46] gower_1.0.0          xfun_0.31            globals_0.14.0
##  [49] rvest_0.3.6          lifecycle_0.2.0      rstatix_0.7.0
##  [52] future_1.21.0        MASS_7.3-53          zoo_1.8-8
##  [55] scales_1.1.1         ipred_0.9-12         hms_1.0.0
##  [58] parallel_4.0.5       yaml_2.2.1           rpart_4.1-15
##  [61] stringi_1.5.3        highr_0.8            foreach_1.5.1
##  [64] e1071_1.7-4          hardhat_0.2.0        boot_1.3-26
##  [67] lava_1.6.10          shape_1.4.5          rlang_0.4.10
##  [70] pkgconfig_2.0.3      evaluate_0.14        labeling_0.4.2
##  [73] recipes_0.2.0        tidyselect_1.1.0     parallelly_1.23.0
##  [76] plyr_1.8.6           magrittr_2.0.1       R6_2.5.0
##  [79] generics_0.1.2       DBI_1.1.1            pillar_1.6.2
##  [82] haven_2.3.1          withr_2.4.1          abind_1.4-5
##  [85] survival_3.2-7       nnet_7.3-15          future.apply_1.7.0
##  [88] modelr_0.1.8         crayon_1.4.0         car_3.0-13
##  [91] utf8_1.1.4           rmarkdown_2.14       GetoptLong_1.0.5
##  [94] readxl_1.3.1         data.table_1.13.6    ModelMetrics_1.2.2.2
##  [97] reprex_1.0.0         digest_0.6.27        glmnet_4.1-4
## [100] stats4_4.0.5         munsell_0.5.0        beeswarm_0.4.0
## [103] vipor_0.4.5
```